

AUTONOMY ARCHITECTURES FOR A CONSTELLATION OF SPACECRAFT

Anthony Barrett

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
phone: +1 818 393-5372, fax: +1 818 393-5244, e-mail: anthony.barrett@jpl.nasa.gov

ABSTRACT

This paper describes three autonomy architectures for a system that continuously plans to control a fleet of spacecraft using collective mission goals instead of goals or command sequences for each spacecraft. A fleet of self-commanding spacecraft would autonomously coordinate itself to satisfy high level science and engineering goals in a changing partially-understood environment – making feasible the operation of tens or even a hundred spacecraft (such as for interferometer or magnetospheric constellation missions).

1. INTRODUCTION

Until the past 5 years, missions typically involved fairly large expensive spacecraft. Such missions have primarily favored using older proven technologies over more recently developed ones, and humans controlled spacecraft by manually generating detailed command sequences with low-level tools and then transmitting the sequences for subsequent execution on a spacecraft controller.

This approach toward controlling a spacecraft has worked spectacularly on previous NASA missions, but it has limitations deriving from communications restrictions – scheduling time to communicate with a particular spacecraft involves competing with other projects due to the limited number of deep space network antennae. This implies that a spacecraft can spend a long time just waiting whenever a command sequence fails. This is one reason why the New Millennium program has an objective to migrate parts of mission control tasks onboard a spacecraft to reduce wait time by making spacecraft more robust [Muscettola et al. 97]. The migrated software is called a “remote agent” and can be partitioned into 4 components:

- a mission manager to generate the high level goals,
- a planner/scheduler to turn goals into activities while reasoning about future expected situations,
- an executive/diagnostician to initiate and maintain activities while interpreting sensed events through reasoning about past and present situations, and
- a conventional real-time subsystem to interface with the spacecraft to implement an activity’s primitive actions.

In addition to needing remote planning and execution for isolated spacecraft, a trend toward multiple-spacecraft missions points to the need for remote distributed planning and execution. The past few years have seen missions with growing numbers of probes. Pathfinder has its rover (Sojourner), Cassini has its lander (Huygens), Cluster II has 4 spacecraft for multi-point magnetosphere plasma measurements. This trend is expected to continue to progressively larger fleets. For example, one proposed interferometer mission [Mettler&Milman 96] would have 18 spacecraft flying in formation in order to detect earth-sized planets orbiting other stars. Another proposed mission involves 5 to 500 spacecraft in Earth orbit to measure global phenomena within the magnetosphere.

To describe the 4 software components of autonomous spacecraft and constellations, the next section describes a master/slave approach toward autonomously controlling constellations. While being a conceptually simple extension to single-spacecraft autonomy, this approach has several problems that motivate the next section on teamwork. Teamwork replaces masters and slaves with leaders and followers, where a follower has the autonomy to look after its teammates. The fourth section discusses ways to expand teamwork to let each spacecraft function both as a leader and a follower, and the last section concludes by discussing hybrids of the three architectures.

2. MASTER/SLAVE COORDINATION

The easiest way to adapt autonomous spacecraft research to controlling constellations involves treating the constellation as a single spacecraft. Here one spacecraft directly controls the others as if they were connected. The controlling “master” spacecraft performs all autonomy reasoning while the slaves only transmit sensor values to the master and forward control signals received from the master to their appropriate local devices (fig 1). The executive/diagnostician starts actions and the master’s real-time subsystem controls the action either locally or remotely through a slave.

The 3 modules above the real-time subsystem essentially follow the standard belief-desire-intention (BDI) framework [Rao&Georgeff 95]. The mission manager takes a set of *beliefs* and generates *desires* (goals) for the

its sensed outcomes, and the constellation's actual state will drift from the expected state and cause future expectations to drift as well. The planner repairs the tasks whenever this drift causes a conflict.

2.3. MISSION MANAGER

This module facilitates high-level spacecraft commanding by maintaining beliefs involving the high-level mission profile. This profile contains a high level behavioral description for the spacecraft. This description can take many forms from a simple set of temporally constrained goals to an elaborate production system that asserts goals upon detecting user specified scientific opportunities by analyzing parts of the constellation & environment model.

For instance, the spacecraft would have periodic goals to transmit data to Earth. These goals would be temporally constrained in order to synchronize with a ground station. They also have to be high level to determine how to communicate based on the specific state of the spacecraft prior to preparing for a downlink. As another example, the mission manager might apply a feature detection algorithm on a previously captured picture and generate observation goals based on the results.

While a spacecraft can operate entirely autonomously with a mission profile. Humans analyzing the science results will tend to suggest changes to mission goals for answering questions arising from their analysis. We can even vary the constellation's level of autonomy by varying the abstractness of the mission profile. When using primitive action sequences, the profile can short-circuit the planner to allow absolute commanding. Adding abstract tasks to the profile lets the spacecraft adapt its behavior to its local environment, and adding data analysis for rule based autonomous goal generation makes a spacecraft detect and respond to scientific opportunities.

3. TEAMWORK

While the master/slave approach benefits from conceptual simplicity, it relies on an assumption that the master spacecraft's real-time subsystem can continuously monitor the slaves' hardware, and this relies on high-bandwidth highly-reliable communications. Since unintended results occur fairly rarely, one way to relax the bandwidth requirements involves putting real-time subsystems on the slaves and only monitoring unexpected events. Unfortunately, this disables the ability to monitor for unexpected events between spacecraft and leads to a host of coordination problems among the slaves [Tambe 97]. Also, failures in the communications system can result in losing slaves.

We can apply teamwork models [Tambe 97, Stone& Veloso 98] to reduce the communications problem by giving the slaves their own executives (fig. 3). This replaces the master/slaves relationship with one between a

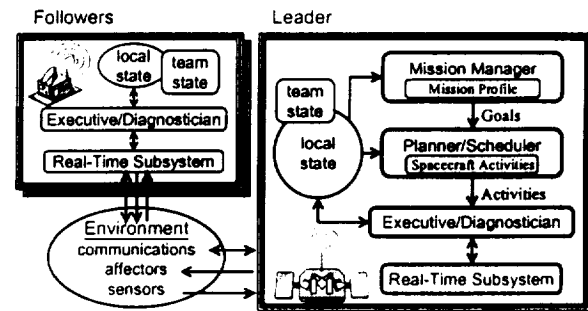


FIG. 3 Architecture for Teamwork

team leader and its followers. Here each follower can monitor its own performance and selectively transmit results to the leader. Partitioning the system's state into local spacecraft states and shared team-states facilitates this selective transmission. While the spacecraft keep their local states private, they communicate to keep team-states consistent across teams in the constellation.

3.1. REPRESENTING TEAM PLANS

Instead of sending separate actions to each follower for execution, the leader broadcasts the entire reactive team plan¹ to all followers. This lets each follower actively monitor its own progress and passively track its teammates' activities. This passive monitoring process maintains robustness while reducing communications.

In addition to regular activities found in the master/slave approach, reactive team plans also include *team activities*. These define coordination points where the team synchronizes before and after executing the team activity. For instance, a 3 spacecraft interferometer has a combiner spacecraft to generate pictures by processing light reflected from two collector spacecraft. A reactive team plan to control the constellation might have 3 team activities (fig. 4) to coordinate the 3 spacecraft while making an observation, and each activity has 2 or 3 sub-activities defining how the constellation behaves during the joint activities. As illustrated, team activities have brackets and those suffixed with an asterisk only apply to subsets of the team. In this case the subset denotes the combiner spacecraft. The activities in this plan subsequently make the constellation attain a rough formation, dress up the formation for finer tolerances to make a measurement, and transmit the results to Earth.

While this interferometer's impoverished number of spacecraft do not sufficiently motivate the need for teamwork, other interferometer mission proposals describe over a dozen, or even a hundred, collectors to support the combiner. To support teamwork for these larger missions,

¹ Given our heavy use of Tambe's formalism, we adopt his terminology and call a sequence a *reactive team plan*.

losing the combiner spacecraft ends the mission anyway, but missions like a 50 satellite constellation are functionally redundant and should not end when any one spacecraft is disabled.

One way to increase robustness involves giving the other spacecraft backup planners and mission managers (fig. 5). While this lets the next spacecraft in a designated chain of command replace a disabled leader, these extra modules are underutilized. Instead of transmitting data to a central spacecraft for planning, we can use the extra planners to move parts of the planning process closer to the data. This makes the spacecraft symmetric and coordination becomes a collaborative effort among peers.

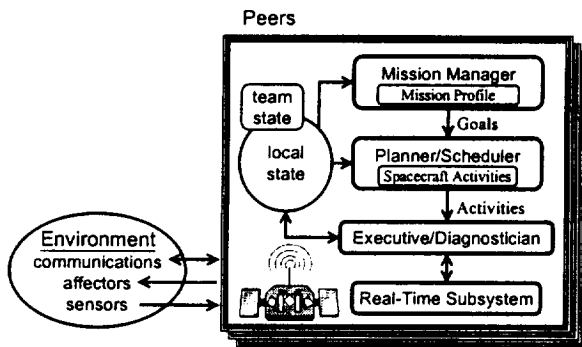


FIG.: 5 Architecture for Peer-To-Peer Coordination

This architecture works particularly well with constellations of satellites that loosely coordinate. For instance, a constellation of picture taking satellites might coordinate to partition desired targets, but each satellite runs in isolation to take its picture. Here the mission managers coordinate to partition the goals, and the planners and executives run in isolation. This class of loose coordination problem is common in the mobile robot community, and some systems even call this module a cooperative planning (or social) module [Müller 96].

4.1. LEVELS OF AUTONOMY

In teamwork or a chain of command, one spacecraft plans how to perform a task and its followers accept and execute the results. Combining loose coordination with teamwork facilitates letting different spacecraft act as leaders for different tasks. Here all spacecraft know about all tasks, and each task has a designated lead spacecraft. Research on autonomy levels [Martin&Barber 96] generalizes this idea. We can give each spacecraft a copy of the plan with tasks annotated with one of 5 autonomy levels:

- Observer: spacecraft does not participate.
- Command-driven: spacecraft serves as a follower.
- Consensus: spacecraft collaboratively plans with others.
- Local: spacecraft plans to perform task alone, and
- Master: spacecraft plans and serves as a leader.

As the 5 definitions imply, autonomy levels specify whether or not a spacecraft can change a task. For instance, a team's leader has tasks annotated with "master", and its followers' tasks have "command-driven" annotations. Given these annotations, a spacecraft can simultaneously serve as a leader and a follower in two separate teams. A spacecraft can even plan and perform tasks in isolation while participating in teams.

While autonomy levels specify which constellation members plan out mission manager requested tasks. These levels are not static – a spacecraft can communicate with the constellation to change a task's autonomy level annotations. For instance, a mission manager might always assign tasks to its spacecraft at the "local" autonomy level. If a team is needed to perform the task, the spacecraft will have to change the annotation to "master." As Martin points out [Martin&Barber 96], this change involves communicating to find spacecraft willing to accept "command-driven" annotations.

Using autonomy levels, we can treat the plan and state information as a shared database where each spacecraft has varying capabilities to modify tasks based on their autonomy-level annotations. Softening the distribution requirement from full to partial plan sharing makes a constellation operate as a team at one point and as multiple independent spacecraft as another. The change involves letting spacecraft keep locally planned and executed tasks private.

4.3. COLLABORATIVE PLANNING

Unlike the other annotations where a single spacecraft plans a task, the "consensus" annotation implies that multiple spacecraft collaboratively plan to perform a task. Collaborative planning involves distributing the plan across the constellation and letting each spacecraft detect and repair problems. The question now becomes a matter of how to keep the plan consistent across the constellation while all spacecraft are updating it. The main objective is to minimize communications overhead while planning.

One approach would fragment the plan and distribute the fragments [Corkill 79]. Since the fragments are disjoint, their union would be consistent. Each spacecraft would expand its own fragment and communicate to detect and resolve interactions. To detect interactions, each spacecraft broadcasts its fragment's effects upon determining them. When a spacecraft hears of an effect that either helps or hinders its own fragment, it initiates a dialog with the broadcasting spacecraft to add signaling actions to their plans to coordinate the interaction. Thus the required bandwidth depends the amount of interaction.

An alternative approach would give every spacecraft a copy of the plan and have them maintain consistency by broadcasting changes as they make them. The main